REMARKS

The Examiner has maintained his rejection of Claims 1, 2, 7-10, 15-18, 23 and 24 under 35 U.S.C. 103(a) as being unpatentable over Howland et al. (U.S. Patent No. 6,018,741) in view of Waldin et al. (U.S. Patent No. 6,651,249). Applicant respectfully disagrees with such rejection, especially in view of the amendments made hereinabove to each of the independent claims. Specifically, applicant has amended each of the independent claims to include the subject matter of dependent Claim 2 et al.

With respect to independent Claims 1, 9 and 17, the Examiner has relied on the following excerpts from Howland to make a prior art showing of applicant's claimed "establishing a hierarchy of lists of attributes, an attribute being comprised of an attribute identifier and an attribute value, the attribute value being comprised of either a list of attributes or a controlling value used by the security scanner program to control an operation of the security scanner program, the list of attributes being comprised of a grouping attribute and a series of one or more attributes" (see this or similar, but not identical, language in each of the foregoing claims).

> "All of the attributes of the root (topmost node) of the tree are locally defined within the root node. **Each descendant node below the root node can include one or more locally defined attributes which override one or more of the attributes of the root node.** Any attribute that is not locally defined within a node is "inherited." The value of an inherited attribute is the value of the corresponding attribute in the nearest ancestor which defines the value of that attribute locally. Each object is virtually defined by its locally defined attributes and the inherited attribute values obtained from the nearest ancestor

> node which lies between that object and the root node." (Col. 2, lines 33-44-emphasis added)
>
> "Each attribute of each of the child nodes is identified as being either a locally defined attribute or an inherited attribute. In FIG. 1A, attributes of each object which are in plain font (no underscores) are locally defined in the object. **For example, root node 10 (object "obj100") has all four attributes,** A-D, locally defined with the values "1" to "4," respectively. Node 20 has attributes A-C locally defined with the values "1" to "3," respectively. Node 21 has the value of attribute D locally defined as the value "3."" (Col. 3, lines 54-64-emphasis added)
>
> "FIG. 4 shows the distinction between Group tasks and Single tasks. **Group tasks may have either Group tasks or Single tasks as descendant tasks. Single tasks have no descendant tasks.** More importantly, Group and Single tasks represent different types of objects." (Col. 6, lines 63-67-emphasis added)

Applicant has amended each of the independent claims to further clarify the claims as follows: "the attribute value being comprised of a list of attributes, the list of attributes being comprised of a grouping attribute and a series of one or more attributes."

Applicant respectfully asserts that the above excerpts from Howland do not teach "the attribute value being comprised of a list of attributes" (emphasis added). Rather, Howland teaches that a "descendent node below the root node can include one or more locally defined attributes" and that "group tasks may have either group tasks or single tasks as descendent tasks..." but "single tasks have no descendent tasks" (see emphasized excerpt above).

Clearly, Howland is teaching parent and child nodes, wherein the nodes may have group tasks (i.e. the node has children - see Fig. 4, item 130) or single tasks (the node does not have children - see Fig. 4, item 120). Further, Howland discloses that a group task "is a concrete instantiation having an object that lists

all descendent tasks in the next level immediately below the group task..." (Col. 7, lines 5-7).

Such a list is not a value, as claimed by applicant, whereby the list is identified by an attribute identifier. In addition, such list does not meet applicant's claimed list, since applicant's list is "comprised of a grouping attribute **and** a series of one or more attributes" (emphasis added), whereas Howland's list is simply comprised of "all descendent tasks in the next level immediately below the group task."

Still yet, the Examiner has relied on the following excerpt, along with Figures 1A and 1B, from Howland to make a prior art showing of applicant's claimed "updating the element and all other elements of the list of attributes if the grouping attribute indicates that updating the element requires all other elements to be updated" and "wherein the grouping attribute is associated with the entire list of attributes for controlling the updating through selection of at least one of at least three scenarios by indicating at least one of" (see this or similar, but not identical, language in each of the foregoing claims).

> "FIG. 1B shows tree 90 after two different changes have been made. First, the attribute values of the root node 10 are changed, so that attributes C and D both have the value "1". Second, node 40 (Object "Obj400") and the subtree which has node 40 as its root are moved to a new parent node 30.
>
> With respect to the first change, the system dynamically assigns, to each inherited attribute of each one of the child nodes (attribute D in node 20 and attribute C in node 21), the current value ("1" and "1", respectively) of the corresponding attributes C and D in the parent node 10. The updated values of attributes C and D are automatically provided in nodes 21 and 20, respectively, when requests for C and D are made after the value of the corresponding attribute of the parent node 10 changes.
>
> On the other hand, attribute C in node 20 and attribute D in

node 21 are each locally defined. **The system keeps the values of each locally defined attribute of the child nodes while the inherited attributes D in node 20 and C in node 21 are being updated.** All of the values of attribute C in the third level of the tree (nodes 30-32) are locally defined, so the change to the root node does not propagate any further down the tree." (Col. 4, lines 15-37-emphasis added)

Applicant respectfully asserts that the above excerpt from Howland teaches only updating inherited values, and not local values (see emphasized excerpt above). Clearly this does not meet applicants claim language since Howland fails to teach "updating the element and all other elements of the list of attributes..." (emphasis added). Thus, applicant does not only disclose "updating the element," which may be the inherited value in Howland as disclosed above, but applicant also claims updating "all other elements of the list of attributes" where, when taken in claimed context, the "list of...attributes [is] for a computing node" such that all attributes in the list of attributes for the computing node are also updated.

Furthermore, the above excerpt from Howland also fails to meet the applicant's remaining claim language, as rejected above, since Howland does not teach "...if the grouping attribute indicates that updating the element requires all other elements to be updated." The above excerpt from Howland does not teach any sort of "grouping attribute" which indicates that updating the element requires all other elements to be updated. The only type of grouping attribute disclosed above is an inherent/local attribute, but neither of those attributes are capable of indicating whether to update an entire list of elements within a node. Instead, the inherent/local attribute simply tells whether to update the inherent attribute only or the local attribute only. Thus, applicant's claim language has clearly not been met by the Howland reference.

For the same reasons just stated, the Howland reference has not met applicant's claimed "wherein the grouping attribute is associated with the entire list of attributes for controlling the updating," since Howland's grouping attribute only relates to either an inherent attribute or a local attribute, and not an entire list of attributes.

In addition, the Examiner has relied on the following excerpt from Howland to meet applicant's claimed "wherein updating the element involves overwriting the value with another value that may be identical to an original value; wherein updating the element and all other elements of the list of attributes involves overwriting each value with another value that may be identical to an original value; wherein updating the element, all other elements in the list of attributes, and all subordinate elements of the list of attributes involves overwriting each value with another value that may be identical to an original value for each element and each subordinate element of the list of attributes" (see this or similar, but not identical, language in each of the foregoing claims).

> "On the other hand, attribute C in node 20 and attribute D in node 21 are each locally defined. The system keeps the values of each locally defined attribute of the child nodes while the inherited attributes D in node 20 and C in node 21 are being updated. All of the values of attribute C in the third level of the tree (nodes 30-32) are locally defined, so the change to the root node does not propagate any further down the tree.
>
> **With respect to the second type of change referred to above, child node 40 is reassigned from its old parent node 31 to a new parent node 30. In response to a request for an inherited attribute A, B, or D of the re-assigned child node 40, the system automatically provides the current values "2," "2," and "2," respectively, of the corresponding attributes A, B, and D in the new parent node 30. (Before the move, node 40 inherited the values "4," "4," and "4" from node 31.)**
>
> The subtree which has node 40 as its root is changed in

> accordance with the new attribute values inherited from new parent node 30. Thus, the inherited values of attributes A, B, and D of node 50 also change to "2," "2," and "2," respectively." (Col. 4, lines 37-50-emphasis added)
>
> "...this request is essentially handled the same way regardless of whether the request comes in any of the following situations:
>
> (1) after the attribute values for the objects in the tree structure are first set, before any of the attribute values are changed;
>
> (2) after one or more of the attribute values of one or more of the objects are changed; or
>
> (3) after one or more of the child nodes is moved within the tree structure 90, by assigning the moved child node to a new parent. (This may be accomplished by modifying the pointer in the child node that points to the parent of the child node.)
>
> At step 214, a determination is made whether the attribute is locally defined in the object (node) at which the attribute is referenced." (Col. 5, lines 1-16)

Applicant respectfully asserts that the above excerpts do not meet applicant's claim language since they completely fail to teach any sort of "overwriting the value with another value that may be identical to an original value" (see this or similar claim language for each of the claim limitations cited above). Specifically, the Examiner has relied Howland's teaching of updating inherited values when a child node is reassigned to a new parent node. However, after carefully reading Howland's disclosure in view of the associated Figures 1A and 1B, it is clear that Howland does NOT teach "overwriting the value with another value that may be identical to an original value," but instead only overwriting values that are different.

For instance, Howland discloses that child node 40 had original values A-D of "4", "4", "3" and "4" respectively when it was assigned to parent node 31 (see Fig. 1A). When child node 40 was assigned to new parent node 30, Howland

specifically discloses that only attributes A, B and D were updated because these are inherited attributes (see emphasized excerpt above). Thus, since attribute C was not an inherited attribute, child node 40 retained the value of "3" for C. Howland simply does not disclose that such value was overwritten even though it remained the same after child node 40 was reassigned to a new parent. Thus, updating "all other elements of the list of attributes" with "another value that may be identical to an original value," as claimed by applicant, are clearly not met by the Howland reference.

Still yet, the Examiner has relied on the following excerpt from Lunt to make a prior art showing of applicant's claimed "wherein the attribute value includes a second list of attributes used by the security scanner program to control the operation of the security scanner program; wherein the second list of attributes includes a second grouping attribute associated with each of the attributes of the second list" (see this or similar, but not identical, language in each of the foregoing claims).

> "An object structure defines a hierarchical grouping of attributes on a business object. The attributes of a business object are often organized into functional groupings that reflect the use of that business object in a particular setting. These groupings can be nested into a hierarchy, with an attribute (and its corresponding attribute descriptor) at each of the "leaves."
>
> An example of an object structure for the business object "Customer" 1 is shown in FIG. 1. Each node 2, 12 represents an attribute on the Customer object 1 with an appropriate attribute descriptor. **The node 2 for "general information" includes groups** for company identification 3, ship-to information 7, and bill-to information 10. The group for company identification 3 includes leaves for customer number 4, customer name 5, and stock identifier 6. The group for ship-to information 7 includes leaves for ship-to address 8 and preferred carrier 9. The group for bill-to information 10 includes a leaf for bill-to address 11.
>
> The node 12 for "accounting" includes groups for credit 13,

billing 16, and bill-to information 18. The group for credit 13 includes leaves for credit limit 14 and credit code 15. The group for billing 16 includes a leaf for AP contact 17. The group for bill-to information 18 includes a leaf for bill-to address 19.

Each of these groups contains attributes as specified in FIG. 1 The attributes that are appropriate for this particular representation are grouped under at least one functional group. This grouping of attributes into a functional-based hierarchy is called an object structure. A business object can have zero to many object structures associated with it.

Each group or attribute in an object hierarchy is called a node." (Col. 4, line 42-Col. 5, line 6-emphasis added)

Applicant respectfully asserts that the above excerpt from Lunt does not meet applicant's claimed "wherein the attribute value includes a second list of attributes..." and "wherein the second list of attributes includes a second grouping attribute associated with each of the attributes of the second list." Specifically, Lunt teaches a node that includes groups (see emphasized excerpt above). Lunt does not even teach a single list of attributes which includes a grouping attribute let alone a "second list of attributes [which] includes a grouping attribute associated with each of the attributes of the second list." Lunt discloses a node 2, which includes groups 3 that are branched off from the node 2, with leaves 4 that further branch off from the group (see Figure 1). Such a hierarchy is system clearly does not meet applicant's claimed "second list of attributes..."wherein the second list of attributes includes a second grouping attribute associated with each of the attributes of the second list" (emphasis added).

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must

be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck*,947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

Applicant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above. Nevertheless, despite such paramount deficiencies and in the spirit of expediting the prosecution of the present application, applicant has substantially incorporated the subject matter of Claim 2 et al. into each of the independent claims.

With respect to dependent Claim 2 et al., presently incorporated in each of the independent Claims, the Examiner has relied on the following excerpt from Howland to make a prior art showing of applicant's claimed "wherein the element of the list of attributes contains an identifier that uniquely identifies the element and a value, wherein the value may itself be a list of elements."

> "Each attribute of each of the child nodes is identified as being either a locally defined attribute or an inherited attribute. In FIG. 1A, attributes of each object which are in plain font (no underscores) are locally defined in the object. For example, **root node 10 (object "obj100") has all four attributes, A-D**, locally defined with the values "1" to "4," respectively. Node 20 has attributes A-C locally defined with the values "1" to "3," respectively. Node 21 has the value of attribute D locally defined as the value "3."" (Col. 3, lines 55-61-emphasis added)

Applicant respectfully asserts that the above excerpt from Howland simply teaches a node with four attributes (see emphasized excerpt above). This, however, fails to meet applicant's specific claim language since it does not teach "the element of the list of attribute contains an identifier that uniquely identifies the element and a value, wherein the value may itself be a list of elements" (emphasis added). Howland simply shows one level of attributes within a node, whereas applicant's claim language clearly includes an "element of the list of attributes" wherein that element also "may itself be a list of attributes," thus providing for multiple levels of lists within lists.

Since at least the third element of the *prima facie* case of obviousness has not been met, a notice of allowance or a specific prior art showing of all of the claim limitations, in the context of the remaining elements, is respectfully requested.

Still yet, applicant brings to the Examiner's attention the subject matter of new Claims 27-28 below, which are added for full consideration:

> "wherein the list of security scanner attributes determines the files to be scanned by the security scanner program" (see Claim 27); and
>
> "wherein the list of security scanner attributes further determines a corrective action to take in response to the identification of an infected file" (see Claim 28).

A notice of allowance or a specific prior art showing of each of the foregoing features, in combination with the remaining claim elements, is respectfully requested.

All of the independent claims are thus deemed allowable for the reasons set forth hereinabove. Moreover, by virtue of their dependence on such claims, all of the remaining dependent claims are also deemed allowable.

A notice of allowance is respectfully requested.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-1351 (Order No. NAI1P251).

Respectfully submitted,
Zilka-Kotab, PC.

Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172-1120
408-505-5100